

# Redirect to WordPress Maintenance Page using .htaccess

blog post: <https://wpmove.net/wordpress/redirect-wordpress-maintenance-htaccess-p913>

When should you do Redirect to the WordPress Maintenance Page? Well, when you're about to do some maintenance work.

From time to time you will need to redirect users to another location or a file that tells the users that you're doing some maintenance work. Of course you can use a WordPress plugin to do that but it's way faster to use .htaccess to redirect the users temporarily to another location and to provide them with some sort of a status update.

Before doing the maintenance it's good to plan for it and put a notice on your WordPress site that on a given date the site will be taken down for maintenance work. We highly recommend that you pick a time when it's not that visited so the effected users are less.

## Why do a Redirect to WordPress Maintenance Page using .htaccess?

Using .htaccess redirect is a great idea because it will save server resources as the PHP engine won't be executed. This is important because if you're doing heavy maintenance work you want every single resource at your disposal to complete the maintenance as quickly as possible.

The idea is to take the entire site down for maintenance and create a file called maintenance. A maintenance file would be displayed when users access the site. Ideally, you may want to whitelist certain IP addresses so you see the site's content and monitor the maintenance progress.

**Before doing anything, backup your whole site using at least 2-3 methods just to be on the safe side.**

The code below should be added at the top of the .htaccess file so they take precedence over any other rules. The file should reside in the root directory of your site.

Before making any changes to the .htaccess file make a copy of it and put the current date to the file

e.g. .htaccess-YYYYMMDD.php because you may not remember what you did and also in case the Apache server doesn't have some modules installed or activated you can revert your changes pretty quickly that way.

After the maintenance work is done it's good to turn off the maintenance mode by putting # back in front of the **APP\_MAINTENANCE\_MODE** row. This will deactivate the whole block after it.

That way you can keep that block of code for future maintenance.

The following rules apply to the Apache web server if using Nginx or a different server you will need those rules adapted for you.

Most of the time you may have noticed that when dealing with **WordPress Migration** most of the redirects are either 301 (permanent redirect) or 302 (temporary redirect) but now we'll have to do a 503 Service Temporarily Unavailable. This is because we want search engines to know that we're doing something temporary such as maintenance work and they should not be indexing the site and its content.

First step is to create a maintenance.html in the root folder of your WordPress site. The root folder is usually called www, public\_html, htdocs, httpsdocs etc. or is usually where your wp-config.php file resides.

Make sure you uncomment #Define APP\_MAINTENANCE\_MODE line so the rules start working.

## **Option 1: Redirect all web traffic to a status page on an external server for more information**

Make sure you change status.example.com with your status page.

```
#####  
# https://WPMove.net - Maintenance .htaccess block  
# Blog post - https://wpmove.net/913  
#Define APP_MAINTENANCE_MODE 1  
<IfDefine APP_MAINTENANCE_MODE>  
    ErrorDocument 503 /maintenance.html  
  
    # Redirect all traffic to an external status page  
    <IfModule mod_rewrite.c>  
        RewriteEngine On  
        RewriteRule . https://status.example.com [R=503,L,QSA]  
    </IfModule>  
</IfDefine>  
# /Maintenance block  
#####
```

## Option 2: Redirect visitor traffic but allow access to assets (images, videos and js/css)

```
#####  
# https://WPMove.net - Maintenance .htaccess block  
# Blog post - https://wpmove.net/913  
#Define APP_MAINTENANCE_MODE 1  
<IfDefine APP_MAINTENANCE_MODE>  
    ErrorDocument 503 /maintenance.html  
  
    # Redirect visitor traffic but allow access to assets (images,  
    videos and js/css)  
    # Blog post - https://wpmove.net/913  
    <IfModule mod_rewrite.c>  
        RewriteEngine On  
        RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f  
        RewriteCond %{REQUEST_URI}  
        !\.(css|js|jpe?g|png|gif|ico|web[pv]) [NC]  
        RewriteRule .* /maintenance.html [R=503,L,QSA]  
    </IfModule>  
</IfDefine>  
# /Maintenance block  
#####
```

## Option 3: Redirect visitor traffic but allow access to assets (images, videos and js/css) and Certain IP addresses

And allow access to certain whitelisted IP addresses to bypass the maintenance page.

```
#####  
# https://WPMove.net - Maintenance .htaccess block  
# Blog post - https://wpmove.net/913  
# Redirect visitor traffic but allow access to assets (images and  
# js/css)  
# allow access to certain whitelisted IP addresses to bypassed the  
# maintenance page.  
#Define APP_MAINTENANCE_MODE 1  
<IfDefine APP_MAINTENANCE_MODE>  
    ErrorDocument 503 /maintenance.html  
  
    <IfModule mod_rewrite.c>  
        RewriteEngine On  
        RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f  
        RewriteCond %{REQUEST_URI} !/maintenance [NC]  
        ##RewriteCond %{REMOTE_ADDR}  
        !^(111\.222\.333\.444|127\.0\.0\.1)  
        RewriteCond %{REQUEST_URI}  
        !\.(css|js|jpe?g|png|gif|ico|web[pv]) [NC]  
        RewriteRule .* /maintenance.html [R=503,L,QSA]  
    </IfModule>  
</IfDefine>  
# /Maintenance block  
#####
```

Read more about Apache mod\_rewrite rules and flags.

Here are rules as images in case you want to include them in a tutorial

```

7
8 #####
9 # https://WPMove.net - Maintenance .htaccess block
0 # Blog post - https://wpmove.net/913
1 #Define APP_MAINTENANCE_MODE 1
2 <IfDefine APP_MAINTENANCE_MODE>
3     ErrorDocument 503 /maintenance.html
4
5     # Redirect all traffic to an external status page
6     <IfModule mod_rewrite.c>
7         RewriteEngine On
8         RewriteRule . https://status.example.com [R=503,L,QSA]
9     </IfModule>
0 </IfDefine>
1 # /Maintenance block
2 #####
3
26 #####
27 # https://WPMove.net - Maintenance .htaccess block
28 # Blog post - https://wpmove.net/913
29 #Define APP_MAINTENANCE_MODE 1
30 <IfDefine APP_MAINTENANCE_MODE>
31     ErrorDocument 503 /maintenance.html
32
33     # Redirect visitor traffic but allow access to assets (images, videos and js/css)
34     # Blog post - https://wpmove.net/913
35     <IfModule mod_rewrite.c>
36         RewriteEngine On
37         RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f
38         RewriteCond %{REQUEST_URI} !\.(css|js|jpe?g|png|gif|ico|web[pv]) [NC]
39         RewriteRule .* /maintenance.html [R=503,L,QSA]
40     </IfModule>
41 </IfDefine>
42 # /Maintenance block
43 #####
44
47 #####
48 # https://WPMove.net - Maintenance .htaccess block
49 # Blog post - https://wpmove.net/913
50 # Redirect visitor traffic but allow access to assets (images and js/css)
51 # allow access to certain whitelisted IP addresses to bypassed the maintenance page.
52 #Define APP_MAINTENANCE_MODE 1
53 <IfDefine APP_MAINTENANCE_MODE>
54     ErrorDocument 503 /maintenance.html
55
56     <IfModule mod_rewrite.c>
57         RewriteEngine On
58         RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f
59         RewriteCond %{REQUEST_URI} !/maintenance [NC]
60         ##RewriteCond %{REMOTE_ADDR} !^(111\.222\.333\.444|127\.0\.0\.1)
61         RewriteCond %{REQUEST_URI} !\.(css|js|jpe?g|png|gif|ico|web[pv]) [NC]
62         RewriteRule .* /maintenance.html [R=503,L,QSA]
63     </IfModule>
64 </IfDefine>
65 # /Maintenance block
66 #####
67

```